REMARKS

In response to the Office Action mailed May 20, 2008, Applicants respectfully request reconsideration. Claims 1-33, 37 and 39 were previously pending in this application. By this amendment, claims 1, 19, 21-26, 30 and 37 have been amended. As a result, claims 1-33, 37 and 39 are pending for examination with claims 1, 19, 21 and 37 being independent. No new matter has been added.

Rejections Under 35 U.S.C. §103. I

The Office Action rejected claims 1-5, 7-15, and 18-20 under 35 U.S.C. §103(a) as allegedly being unpatentable over U.S. Patent No. 5,826,014 ("Coley") in view of U.S. Patent No. 6,233,688 ("Montenegro"). Applicants respectfully disagree.

A. Independent Claim 1

Claim 1, as amended, recites:

> A computer-implemented method, comprising:
> receiving, by an operating system and/or an enforcement module which is associated with or is part of the operating system, *a call from a firewall aware application via a first application programming interface*, the call having parameters for a connection to an endpoint that the firewall aware application desires to establish, whereby the firewall aware application explicitly communicates a request to traverse a firewall to establish the connection, *the request is being directed to a specific socket and includes handling requirements for data sent and/or received by the firewall aware application*; and
> making, by the operating system and/or the enforcement module, *a call via a second application programming interface* to the firewall to establish the connection in accordance with the parameters.
> (Emphasis added).

Claim 1 has been amended to recite, inter alia, receiving, by an operating system and/or an enforcement module which is associated with or is part of the operating system, a call from a firewall aware application via a first application programming interface, the call having parameters for a connection to an endpoint that the firewall aware application desires to establish, whereby the application explicitly communicates a request to traverse a firewall to establish the connection, *the*

request is being directed to a specific socket and includes handling requirements for data sent and/or received by the firewall aware application. (Emphasis added). Support for these amendments can be found, for example, on page 5, paragraph [0009], pages 18-19, paragraph [0040] and page 19, paragraph [0041] of Applicants' specification. In contrast, Coley does not teach or suggest that the request is being directed to a specific socket and includes handling requirements for data sent and/or received by the firewall aware application. In fact, nowhere does Coley even mention socket(s).

On page 7, the Office Action concedes that Coley "does not explicitly teach an application programming interface." The Office Action then states that Montenegro "teaches a firewall traversal method regarding loading an application programming interface (API) onto the client system" in col. 7, lines 17-20. Applicants note that claim 1 recites, inter alia, receiving, by an operating system and/or an enforcement module which is associated with or is part of the operating system, a call from a firewall aware application via *a first application programming interface*, ... making, by the operating system and/or the enforcement module, a call via **a** *second application programming interface* to the firewall to establish the connection in accordance with the parameters. (Emphasis added). Thus, claim 1 recites two types of application programming interfaces.

Further, Montenegro is directed to separating application from firewall traversal mechanisms. (Montenegro, col. 1, lines 58-60). Montenegro makes transparent the process of firewall traversal. (Montenegro, col. 4, lines 1-2). The Remote Access Firewall Traversal Uniform Resource Locator (RAFT URL) of Montenegro provides a universal language that aids in unifying firewall technology. (Montenegro, col. 4, lines 34-35). Further, in the cited passage, Montenegro discusses that if the specific "raft-type" (type of firewall traversal and/or remote access security) is not provided for in the socket factory, by way of methods, functions, classes and/or code that can be executed, then the required methods, functions, classes, code, etc. must be obtained. (Montenegro, col. 7, lines 11-15). In that case, the firewall traversal procedure first gets the needed methods, classes, etc. for the raft-type. (Montenegro, col. 7, lines 15-17). These methods, classes, etc. may be *obtained from the firewall itself* and *then loaded* as an API (Application Program Interface) or mobile code, such as Java applet, *onto the client system.* (Montenegro, col. 7, lines 17-20). (Emphasis added). Thus, the API of Montenegro comprises "methods, classes, etc." obtained from

the firewall itself for the raft-type and then loaded into the client system. Therefore, it appears that this API of Montenegro is different from a first application programming interface recited in claim 1. Indeed, nowhere does Montenegro that this API is used for "receiving, by an operating system and/or an enforcement module which is associated with or is part of the operating system, a call from a firewall aware application," as recited in claim 1. Further, the API of Montenegro is different from a second application programming interface recited in claim 1 since the API of Montenegro is not used to make a call to the firewall to establish the connection in accordance with the parameters for a connection to an endpoint, as recited in claim 1.

In addition, Montenegro discusses that the client application 310 makes use of a socket factory 320 (Application Program Interface) to access a system resource such as sockets. (Montenegro, col. 4, lines 50-52). *The socket factory recognizes the RAFT URL* and *configures itself* 330 to communicate with gateway/firewall 350. (Montenegro, Fig. 5; col. 4, lines 60-62). When provided, the RAFT URL is passed to the socket factory (in Java) (step 530) *that will execute methods needed to perform the firewall traversal procedure*. (Montenegro, Fig. 5; col. 7, lines 6-9). Therefore, the socket factory of Montenegro is also different from the first and second application programming interfaces recited in claim 1.

In view of the above, Montenegro fails to cure the deficiency of Coley and does not teach or suggest "receiving, by an operating system and/or an enforcement module which is associated with or is part of the operating system, a call from a firewall aware application via a first application programming interface, the call having parameters for a connection to an endpoint that the firewall aware application desires to establish, whereby the firewall aware application explicitly communicates a request to traverse a firewall to establish the connection, the request is being directed to a specific socket and includes handling requirements for data sent and/or received by the firewall aware application; and making, by the operating system and/or the enforcement module, a call via a second application programming interface to the firewall to establish the connection in accordance with the parameters," as recited in claim 1.

The Office Action Fails to Establish a *Prima Facie* Case of Obviousness

MPEP §2143 lists several examples of rationales that may be used to establish a *prima facie* case of obviousness (i.e., combining prior art elements according to known methods to yield predicable results, simple substitution of one known element for another to obtain predictable results, etc.). As a preliminary matter, Applicants respectfully note that the Office Action fails to explicitly indicate any of the rationale set forth in MPEP §2143. While not clear from the text of the Office Action, the Office Action appears to rely on the first rationale, "(A) combining prior art elements according to known methods to yield predicable results," in support of the assertion that independent claim 1 is purportedly rendered obvious by Coley and Montenegro.

MPEP §2143(A) describes in detail the requirements for rejecting a claim based on this rationale. This section of the MPEP states that the Office Action **must** articulate, *inter alia*, the following: (1) a finding that the prior art included each element claimed, although not necessarily in a single prior art reference, **with the only difference between the claimed invention and the prior art being the lack of actual combination of the elements** in a single prior art reference. (Emphasis added).

The Office Action has failed to establish a *prima facie* case of obviousness, as the first requirement set forth in MPEP §2143(A) clearly is not met. Specifically, the Office Action fails to establish that the prior art includes each claimed element recited in independent claim 1. For example, as discussed in above, neither Coley nor Montenegro discloses or suggests a first and a second application programming interfaces recited in claim 1.

MPEP §2143(A) further requires that the Office Action articulate a finding that one of ordinary skill in the art could have combined the elements as claimed by known methods, and with no change in their respective functions. The Office Action also must articulate a further finding that the results of the combination would have been recognized as predicable to one of ordinary skill in the art.

Even if Coley and Montenegro taught all claimed elements, which they do not, the Office Action does not provide any rationale for combining elements from these references; in particular, the Office Action fails to articulate the further findings indicated above as required by MPEP §2143(A).

Instead, the Office Action merely asserts on page 7 that "it would have been obvious to one of ordinary skill in the art at the time of the invention to implement an application programming interface as taught by Montenegro. The suggestion/motivation for implementing an application programming interface would have been to provide an interface between applications and the operating system achieving a form of standardization for applications to interface with the operating system." Thus, the Office Action does not state any rationale for combining elements from Coley and Montenegro. It is not clear from the above statements of the Office Action in what way "implementing an application programming interface as taught by Montenegro" results in limitations recited in claim 1.

Coley describes a firewall operating on a stand alone computer. (Coley, Abstract). *The firewall application* running on the stand alone computer *includes a variety of proxy agents* that are specifically assigned to an incoming request in accordance with the service protocol (i.e., port number) indicated in the incoming request. (Coley, Abstract). (Emphasis added). In a preferred embodiment of Coley, *a proxy agent is assigned to a request* based on the service associated with an access request (e.g., the Telnet port number is indicated). (Coley, col. 6, lines 21-23). Therefore, different proxy agents may be assigned to different access requests. Coley does not suggest providing "an interface between applications and the operating system achieving a form of standardization for applications to interface with the operating system," as stated in the Office Action.

Furthermore, while Coley describes a firewall operating *on a stand alone computer*, Montenegro teaches that the RAFT URL may be provided *to any client application.* The socket factory (Application Program Interface) of Montenegro is configured to understand the naming convention (RAFT URL) and then initiate steps to obtain remote access, which includes the negotiation of security protocols defined by the RAFT URL. (Montenegro, col. 4, lines 55-58). The RAFT URL is input either by user or by preference setting to the client application 310. (Montenegro, col. 4, lines 58-60). *The socket factory recognizes the RAFT URL* and *configures itself* 330 to communicate with gateway/firewall 350. (Montenegro, Fig. 5; col. 4, lines 60-62). Montenegro also describes, with reference to Fig. 6, that once the appropriate RAFT URL is provided to remote node 610 the RAFT URL is passed to a *socket factory generated by some*

application running in memory 614 and executed by processor 612. (Montenegro, col. 8, lines 46-50).

Further, Montenegro discusses that client application 310 (or 312 or 315) would be *unaware*, except for the obtaining of behavior from the sockets *of the firewall traversal.* (Montenegro, col. 5, lines 9-11). (Emphasis added). The traversing of the firewall is divorced from the client application 310 and made the responsibility of the socket factory 320. (Montenegro, col. 5, lines 15-17). In so doing, traversal of the firewall and its security method is made *transparent* to client applications 310, 312 and 315. (Montenegro, col. 5, lines 17-19). (Emphasis added). In contrast, claim 1 recites receiving, by an operating system and/or an enforcement module which is associated with or is part of the operating system, a call from *a firewall aware* application via a first application programming interface, *the call having parameters for a connection to an endpoint that the firewall aware application desires to establish*, whereby the firewall aware application *explicitly communicates a request to traverse a firewall* to establish the connection. (Emphasis added).

Thus, Coley and Montenegro describe different methods of firewall traversal. As conceded in the Office Action, Coley "does not explicitly teach an application programming interface." Montenegro, as discussed above, describes the socket factory (Application Program Interface) and methods, functions, classes, code, etc. for the "ratf-type" processing that can be obtained from the firewall and loaded as an API or mobile code onto the client system. Therefore, it is not clear how one of ordinary skill in the art would combine either the socket factory or the "methods, functions, classes, code, etc." for the for the "ratf-type" processing taught by Montenegro with a stand alone firewall of Coley including proxy agents each assigned to a specific request to obtain "receiving, by an operating system and/or an enforcement module which is associated with or is part of the operating system, a call from a firewall aware application via a first application programming interface, the call having parameters for a connection to an endpoint that the firewall aware application desires to establish, whereby the firewall aware application explicitly communicates a request to traverse a firewall to establish the connection, the request is being directed to a specific socket and includes handling requirements for data sent and/or received by the firewall aware application; and making, by the operating system and/or the enforcement module, a call via a

second application programming interface to the firewall to establish the connection in accordance with the parameters," as recited in claim 1.

In the view of the foregoing, there is no rationale for combining elements from Coley and Montenegro. Accordingly, the Office Action has failed to establish a *prima facie* case of obviousness.

In view of the above, claim 1 patentably distinguishes over Coley and Montenegro, either alone or in combination.

Claims 2-18 depend from claim 1 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 1-18 is respectfully requested.

B. Independent Claim 19

Claim 19, as amended, recites:

> A computer system comprising:
> an operating system;
> *a first application programming interface associated with the operating system and configured and adapted to receive a call from a firewall aware application*, the call having parameters for a connection to an endpoint that the firewall aware application desires to establish, whereby *the firewall aware application explicitly communicates a request to traverse a firewall to establish the connection, the request is being directed to a specific socket and includes handling requirements for data sent and/or received by the firewall aware application*; and
> an enforcement module associated with or is part of the operating system and called via the application programming interface and configured and adapted to:
> receive an indication from the application that the application desires to establish the connection; and
> *make a call via a second application programming interface to a firewall to establish the connection in accordance with the parameters.*
> (Emphasis added).

With respect to claim 19, on page 10, the Office Action rejected claim 19 "because it is directed to the same subject matter as claim 1." Applicants have amended claim 19 to recite that the request is being directed to a specific socket and includes handling requirements for data sent and/or received by the firewall aware application.

As should be clear from the above discussion, neither Coley nor Montenegro teaches or suggest all limitations of claim 19. Moreover, the Office Action has failed to establish a *prima facie* case of obviousness while rejecting claim 19 as well.

In view of the foregoing, claim 19 patentably distinguishes over Coley and Montenegro, either alone or in combination.

Claim 20 depends from claim 19 and is allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 19 and 20 is respectfully requested.

Rejections Under 35 U.S.C. §103. II

The Office Action rejected claims 21-26, 30, 33 and 37 under 35 U.S.C. §103(a) as allegedly being unpatentable over U.S. Patent No. 7,146,638 ("Malcolm") in view of Montenegro. Applicants respectfully disagree. In addition, without acceding to the appropriateness of the rejection, Applicants have amended independent claims 21 and 37 to more clearly distinguish over the cited references. Support for these amendments can be found, for example, on page 26-27, paragraph [0059], pages 33-34, paragraphs [0073]-[0074] and pages 35-36, paragraphs [0077]-[0079] of Applicants' specification.

A. Independent Claim 21

Claim 21, as amended, recites:

> A computer-implemented method, comprising:
> *receiving, by an interception module communicating with a firewall via a first application programming interface, via a second application programming interface at least one policy established by a first user that permits at least one of an application and a service to connect to a network when the first user runs the at least one of the application and a service*, wherein the at least one policy is stored among a plurality of policies in a policy cache of the interception module;
> receiving, by the interception module a connect attempt, a listen attempt, or a combination thereof from the application or the service run by a second user;
> extracting, by the interception module, user and application or service information from the connect attempt, the listen attempt, or the combination thereof;
> *determining, by the interception module, an identity of the second user and what application or what service is making the connect attempt, the listen attempt, or the combination thereof*;

> *determining, by the interception module, whether the identity of the second user matches an identity of a user that established the at least one policy* and whether the connect attempt, the listen attempt, or the combination thereof comply with the at least one policy; and
>
> *when the connect attempt, the listen attempt, or the combination thereof comply with the at least one policy and the identity of the second user matches the identity of the user that established the at least one policy*, instructing, by the interception module, the firewall to automatically create a configuration to allow the connect attempt, the listen attempt, or the combination thereof, and storing the configuration in a filter cache of the interception module.
> (Emphasis added).

On pages 10 and 11, the Office Action states that Malcolm teaches some limitations of claim 21. Applicants have amended independent claim 21 to recite, inter alia, "receiving, by an interception module communicating with a firewall via a first application programming interface, via a second application programming interface at least one policy established by a first user that permits at least one of an application and a service to connect to a network when the first user runs the at least one of the application and a service," "determining, by the interception module, whether the identity of the second user matches an identity of a user that established the at least one policy and whether the connect attempt, the listen attempt, or the combination thereof comply with the at least one policy."

Malcolm discusses a firewall that is responsible for intercepting an access request directed from the application program to a destination address on the wide area network, identifying one of the at least one access request definitions that matches the intercepted access request, and *prompting a user to approve or deny the intercepted access request* accompanied by the justification statement from the identified access request definition. (Malcolm, col. 4, lines 12-19). (Emphasis added). After informing the user about the access request, the firewall receives a user response indicating approval or denial of the intercepted access request. (Malcolm, col. 4, lines 38-40). Furthermore, the firewall program may maintain an access rule data structure and prompt the user to provide an instruction whether to apply the user response against subsequent access requests matching the identified access request definition. (Malcolm, col. 4, lines 40-44). Access rules comprise three parameters: application name, destination address or URL, and port, as well as an instruction to

accept or deny the access. (Malcolm, col. 4, lines 47-50). When an access request is approved, either by an access rule or by the user response, the firewall passes the approved access requests to the wide area network. (Malcolm, col. 4, lines 56-59). Malcolm does not teach or suggest limitations of amended claim 21. In particular, Malcolm does not teach or suggest determining, by the interception module, *whether the identity of the second user matches an identity of a user that established the at least one policy* and whether the connect attempt, the listen attempt, or the combination thereof comply with the at least one policy; and *when the connect attempt, the listen attempt, or the combination thereof comply with the at least one policy and the identity of the second user matches the identity of the user that established the at least one policy*, instructing, by the interception module, the firewall to automatically create a configuration to allow the connect attempt, the listen attempt, or the combination thereof, and storing the configuration in a filter cache of the interception module, as recited in claim 21.

On page 12, the Office Action concedes that Malcolm "does not explicitly teach an application programming interface." The Office Action then states that Montenegro "teaches a firewall traversal method regarding loading an application programming interface (API) onto the client system" in col. 7, lines 17-20. Applicants note that claim 21 recites, inter alia, *receiving*, by an interception module *communicating with a firewall via a first application programming interface, via a second application programming interface at least one policy established by a first user*. (Emphasis added). Thus, claim 21 recites two types of application programming interfaces. Furthermore, APIs discussed in Montenegro are different from application programming interfaces recited in claim 21 and Montenegro therefore fails to cure the deficiency of Malcolm. Moreover, as discussed above, Malcolm does not teach or suggest limitations of amended claim 21.

In view of the foregoing, claim 21 patentably distinguishes over Malcolm and Montenegro, either alone or in combination.

Claims 22-33 depend from claim 21 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 21-33 is respectfully requested.

B. Independent Claim 37

Claim 37, as amended, recites:

A computer system, comprising:

a firewall; and

an interception module communicating with the firewall via a first application programming interface, the interception module including a second application programming interface for establishing, by a first user, at least one policy that permits at least one of an application and a service to connect to a network when the first user runs the at least one of the application and a service, wherein the at least one policy is stored in a policy cache of the interception module, the interception module is configured and adapted to:

intercept a request for a connect attempt, a listen attempt, or a combination thereof from the application or the service run by a second user;

extract user and application or service information from the connect attempt, the listen attempt, or the combination thereof;

identify the user and the application or the service from the user and application or service information;

determine whether an identity of the second user matches an identity of a user that established the at least one policy and whether the connect attempt, the listen attempt, or the combination thereof comply with the at least one policy; and

when the connect attempt, the listen attempt, or the combination thereof comply with the at least one policy and the identity of the second user matches the identity of the user that established the at least one policy, instructing the firewall to create a configuration to allow the connect attempt, the listen attempt, or the combination thereof, and storing the configuration in a filter cache of the interception module.

(Emphasis added).

On page 13, the Office Action rejected independent claim 37 "as directed to the same subject matter as claim 21." Claim 37 has been amended similarly to independent claim 21. As should be clear from the above discussion, neither Malcolm nor Montenegro teaches or suggest all limitations of claim 37.

In view of the foregoing, claim 37 patentably distinguishes over Malcolm and Montenegro, either alone or in combination.

Claims 38-39 depend from claim 37 and are allowable for at least the same reasons.

Accordingly, withdrawal of the rejection of claims 37-39 is respectfully requested.

CONCLUSION

A Notice of Allowance is respectfully requested. The Examiner is requested to call the undersigned at the telephone number listed below if this communication does not place the case in condition for allowance.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this response, including an extension fee, the Director is hereby authorized to charge any deficiency or credit any overpayment in the fees filed, asserted to be filed or which should have been filed herewith to our Deposit Account No. 23/2825, under Docket No. M1103.70154US00.

Dated: August 20, 2008

Respectfully submitted,

By _____
James H. Morris
Registration No.: 34,681
WOLF, GREENFIELD & SACKS, P.C.
Federal Reserve Plaza
600 Atlantic Avenue
Boston, Massachusetts 02210-2206
617.646.8000